

<https://helda.helsinki.fi>

---

## Improved behavioral analysis of fuzzy cognitive map models

Hatwagner, M.F.

Springer  
2018

---

Hatwagner , M F , Vastag , G , Niskanen , V A & Kóczy , L T 2018 , Improved behavioral analysis of fuzzy cognitive map models . in L Rutkowski , R Scherer , M Korytkowski , W Pedrycz , R Tadeusiewicz & J M Zurada (eds) , Artificial Intelligence and Soft Computing 1 : 7th International Conference, ICAISC 2018, Zakopane, Poland, June 3-7, 2018, Proceedings, Part II . Lecture Notes in Computer Science , vol. 10842 , Springer , Cham , pp. 630-641 , International Conference on Artificial Intelligence and Soft Computing , Zakopane , Poland , 03/06/2018 . [https://doi.org/10.1007/978-3-319-91262-2\\_55](https://doi.org/10.1007/978-3-319-91262-2_55)

---

<http://hdl.handle.net/10138/310423>

[https://doi.org/10.1007/978-3-319-91262-2\\_55](https://doi.org/10.1007/978-3-319-91262-2_55)

---

unspecified

acceptedVersion

---

*Downloaded from Helda, University of Helsinki institutional repository.*

*This is an electronic reprint of the original article.*

*This reprint may differ from the original in pagination and typographic detail.*

*Please cite the original version.*

# Improved Behavioral Analysis of Fuzzy Cognitive Map Models

Miklós F. Hatwagner<sup>1</sup>, Gyula Vastag<sup>2</sup>, Vesa A. Niskanen<sup>3</sup>, and László T. Kóczy<sup>4</sup>

<sup>1</sup> Dept. of Information Technology, Széchenyi István University, Győr, Hungary,  
`miklos.hatwagner@sze.hu`

<sup>2</sup> Dept. of Leadership and Organizational Communication, Széchenyi István  
University, Győr, Hungary,  
`vastag.gyula@sze.hu`

<sup>3</sup> Dept. of Economics & Management, University of Helsinki, Helsinki, Finland,  
`vesa.a.niskanen@helsinki.fi`

<sup>4</sup> Dept. of Information Technology, Széchenyi István University, Győr, Hungary and  
Dept. of Telecommunications and Media Informatics, Budapest University of  
Technology and Economics, Hungary,  
`koczy@sze.hu`, `koczy@tmit.bme.hu`

**Abstract.** Fuzzy Cognitive Maps (FCMs) are widely applied to describe the internal connections, including their direction and strength among major components of complex systems. The popularity of FCMs application in various fields (e.g. medicine, sustainable waste management) is mostly based on their simple, clear way of system representation, easy model creation and usage for the experts of the field, decision support by simulations using several interesting scenarios, etc.

If historical, measured data of the investigated system is available then the model can be created by using a suitable learning technique. This is the preferable way of model construction, because it is not affected by more or less subjective opinions of human experts. Such data is not always available, however. In these cases the experts involved in model creation have to take the responsibility for the realistic mapping of the system. Unfortunately a small change in the estimated strength of connection between two factors of the system may lead to significantly different simulation outcome. The preliminary exploration of model ‘sensitivity’ to such subtle modifications is very important to decision makers. In these cases specific connections can be further investigated and if necessary, their strength is corrected.

This paper deals with the advanced version of a behavioral analysis. Based on the experiences of the authors, their method is further improved to generate more life-like, slightly modified model versions based on the original one suggested by experts. The details of the method is described, its application and the results are presented by an example of a banking application.

**Keywords:** banking, Fuzzy Cognitive Maps, model uncertainty, multi-objective optimization, Bacterial Evolutionary Algorithm.

## 1 Introduction

The task of well considered decision making may be really hard, and the consequences of a wrong intervention are often serious, especially in an environment where several important, interrelated factors have to be taken into account. According to this, decision support is in the focus of researchers for a long time, and various methods were suggested [1].

This paper deals with Fuzzy Cognitive Maps (FCMs). FCM is a bipolar fuzzy graph: its nodes represent the major components of the modeled system and the arcs among them express the direction and strength of relationships. It describes the operation of a system qualitatively [2] and can be used for decision support [3]. The main advantages of applying FCM are e.g. transparency, ease of use, can be used to model even complex systems.

The FCM model of a system can be created two main ways [4]: the first is based on the knowledge, experiences and competence of one or more experts. The cooperation of multiple experts help to decrease the influence of personal beliefs and subjectivity, but even if the developed model is free from these effects it can be inaccurate. For example, if a model contains only 10 nodes, the number of relationships can be up to 90, and it is often hard to define the strength of so many relations with the required accuracy. That is why the recommended way of model creation is based on historical, measured data and a suitable machine learning technique. These data are sometimes not available, however, and only the expert-based methods can be applied. Unfortunately even a subtle change in connection strengths may change the behavior of the model, e.g. the final, stable states of two slightly different systems can be different despite the same initial state, or the number of possible final states may change. It worth analyze the effect of uncertainty on model behavior before decision making. This work has already begun [5, 6], but the authors improved the method based on their experiences.

The analysis is based on the systematic and automated modification of the strength of relationships. Every modified model version is tested with a predefined huge set of initial states, and the result of simulations are collected and analyzed. The goal of the investigation is to find a slightly modified model that has different or more final stable states, repeats a series of states or behaves chaotically more often. The differentiation of the last two cases is one of the new features of the improved method. The behavioral properties are very interesting for the decision makers. The search was performed by a multi-objective optimization, and the fitness of modified models was defined by a weighted sum. This approach has its disadvantages [7], thus the fitness of models are now expressed on the basis of their Pareto-optimality. The method was already able to find model versions with significantly different behavior, but in order to achieve its goal, it usually had to drastically modify the internal relationships of the model. The improved method strives for more similar original and modified models, because similarity has become one of the optimization targets. Furthermore, the effect of the user-defined  $\lambda$  parameter of FCM's threshold function is also inves-

tigated. It is already known that its value has an effect on the behavior of the model [8].

Section 2 describes briefly the theoretical basics of the applied methods, including FCM and Bacterial Evolutionary Algorithm (BEA). Section 2.3 specifies the interesting details of the implemented program. In order to demonstrate the capabilities of the improved method, a case study is provided in Section 3. Section 4 concludes the results and show the possible ways of further research.

## 2 The applied methods of behavioral analysis

### 2.1 Fuzzy Cognitive Maps

Cognitive Maps were suggested by Axelrod [9] to describe the cause-effect relations of political groups and their possible acts. His technique was further improved by Kosko [10]: the edges of the graph are weighted to express the strength of relations, and also the nodes have numerically defined status. Formally, an FCM can be defined by a 4-tuple:  $(C, W, A, f)$ , where  $C = C_1, C_2, \dots, C_N$  is the set of nodes, called *concepts* in FCM terminology.  $N$  is the number of concepts. Concepts represent the main factors, components of a system or a variable. The status of concept  $i$  at time  $t$  ( $t = 1, 2, \dots, T$ ) is expressed by the *activation value*  $A_i \in \mathbb{R}$ . The function  $A : (C_i) \rightarrow A_i$  associates the activation value to the node. The function  $W : (C_i, C_j) \rightarrow w_{ij}$  defines the *weight* (causal value) of the directed arc between concepts  $C_i$  and  $C_j$ . The weight values are represented with the *connection matrix*. In our paper FCM of Type I [11] is used, where concepts never influence themselves (the main diagonal contains zeros). The weight must fall in the  $w_{ij} \in [-1, +1]$  interval. The last component of the tuple is the *transformation* or *threshold* function  $f : \mathbb{R} \rightarrow [0, 1]$ . This function guarantees that the activation values will remain in their allowed interval during simulations. (In some rare cases, the  $A_i \in [-1, +1]$  can also be used with a matching threshold function.) Several threshold functions were suggested in the literature [12], but only the most common sigmoid function (1) is used in this paper.

The  $\lambda > 0$  parameter defines the steepness of the function, and it is not directly connected to any physically observable properties of the modeled system. Its usual value is 5. With lower  $\lambda$  values the function approximates a linear function, with higher values the sign function.

The activation values are updated by (2) in our case during the consecutive time steps.

$$f(x) = \frac{1}{1 + e^{-\lambda x}} \quad (1)$$

$$A_i^{t+1} = f \left( \sum_{j=1}^N w_{ij} A_j^t \right) \quad (2)$$

A model using continuous activation values can behave three different ways [12] during simulation: i) in most cases it converges quickly to an equilibrium

point, often called *fixed point attractor* (FP). ii) Sometimes a series of activation vectors appears repeatedly always in the same order. This infinite transition among states is called *limit cycle* (LC). iii) If the model behaves *chaotically*, the state of the model never stabilizes.

## 2.2 Bacterial Evolutionary Algorithm

Bacterial Evolutionary Algorithm (BEA) is a member of evolutionary algorithms, capable to solve even non-continuous, non-linear, multi-modal, high dimensional, global optimization problems, and provides the near-optimal solutions of them. Nawa and Furuhashi suggested [13, 14] this straightforward and robust method for the optimization of fuzzy systems' parameters, but it can be successfully applied to other problems as well.

The algorithm works with a collection of possible solutions, called population. The elements of the population often called bacteria as well, because the method imitates the evolution of bacteria in nature. Several generations of the population are generated using the two main operators, *bacterial mutation* and *gene transfer*, until one of the stop conditions (e.g. stopped convergence, limit on time or number of generations) are fulfilled. The best bacteria of the final population are considered as result.

Bacterial mutation explores the search space by random modification of bacteria. The bacteria are mutated individually and independently. First, the copies of an original bacterium, the so-called clones are created. Then the operator iterates over every genes of the bacterium in random order. In every iterative step, the current gene is randomly modified in the clones, and they are evaluated. If the modification leads to better objective value, the new allele is kept and copied to both original and clone bacteria. This technique preserves the old alleles if they serves the goals of optimization better, and explicit elitism is not needed.

Gene transfer exploits and combines the genetic information coded in the bacteria of the current population in order to find even better solutions. At first, it sorts the population based on the objective values of bacteria. Then it divides the population to two halves: the sub-population containing better bacteria is called the superior half, while the other is the inferior half. The operator chooses a bacterium randomly from the superior half, and an other from the inferior half. Next, at least one allele is copied from the better bacterium into the other. The modified bacterium have to be re-evaluated, and if it became better, it has the chance to migrate into the superior half, and scatter its genetic code among other bacteria during the consecutive gene transfers.

## 2.3 Interesting details of the program developed for FCM analysis

The goal of this study was to find a slightly modified model, that behaves radically different than the original model. This way the connections effecting the strongest influence on the behavior of the model (e.g. the simulations lead to more FPs, LCs or chaotic behavior) can be discovered and their values can be further analyzed before using the model for decision support.

The weights in the connection matrix are given by real values in the allowed interval, thus the original search problem defines an infinitely large search space. Because only some of the possible values are used in practical applications (according to the applied linguistic variables), only 9 different weight values are used in our program  $(-1, -0.75, -0.5, \dots, +1)$ . The lack of causal relation between two concepts can be identified by experts with high confidence, therefore the program never changes the zero weight connections of the original model.

The search for modified models is directed by BEA. A bacterium encodes a possible  $\lambda$  value ( $0.1 < \lambda < 10.0$ ) and the new weights of the originally non-zero weight connections. In our case study, the model contains 13 concepts, thus the number of connections can be up to  $12 \times 13 = 156$ . Luckily many possible connections do not exist, but including the  $\lambda$  parameter, they still lead to a 62 variable optimization problem.

The behavior of modified models are tested by simulations. Similarly to connections weights, the set of possible activation values are limited to 0, 0.25, 0.5, 0.75 and 1. The program starts with the generation of 1000 random initial state vectors (scenarios) and all modified models are tested by using the same set of scenarios. The program automatically detects FPs and counts the initial state vectors leading to the same FP. Simulations consist of at most 100 time steps. (State vectors usually converge quickly to an equilibrium state.) If the state vector of concept values stabilizes earlier, it is considered a FP, otherwise the program starts to find a LC. If a repeated sequence of state vectors is not found, it is considered a chaotic behavior.

The state of the system is considered stable only if the values of all concepts has changed by at most 0.001 during the last five consecutive time steps. Unfortunately, the resulting stable states are often not exactly the same even if they can be considered the same in practice, e.g. rounding errors of floating point arithmetic may distort the results. The program creates clusters of final state vectors using k-means clustering [15], and finally these clusters are considered the 'real' FPs. The number of clusters is estimated by gap statistics [16].

The goals of the optimization are the following: i) maximize the number of FPs, ii) maximize the number of LCs, iii) maximize the number of chaotic behavior, iv) minimize the  $d$  difference of modified and original matrix calculated by (3), where  $N$  is the number of concepts,  $o$  is the connection matrix of the original, and  $m$  is the connection matrix of the modified model.

$$d = \sum_{i=1}^N \sum_{j=1}^N (o_{ij} - m_{ij})^2 \quad (3)$$

This multi-objective optimization problem is solved by BEA in a Pareto-optimal manner. The bacteria of a population is classified into several sets: the bacteria on the Pareto-front are collected in the set of the 'first' Pareto-front. The Pareto-front of the other, remaining bacteria can also be determined in the same way, and these bacteria will be in the 'second' Pareto-front, etc. The mutation operator is slightly modified in our program: the Pareto-fronts of the sub-population of a bacterium and its clones are detected, and if the original

bacterium is not an element of the 'first' Pareto-front, its gene is modified to the allele of the first bacterium of the 'first' Pareto-front. The gene transfer operator is also modified in a similar way. The population is sorted on the basis of the Pareto-front number of bacteria.

The population of our example contained 10 bacteria, 3 clones were created for each bacteria during mutation, 3 infections were made in every generation and the optimization stopped after the 10<sup>th</sup> generation.

### 3 Case study: a banking application

The application of the proposed method is demonstrated with a real-life problem. Table 1 contains the description of major concepts of a specific bank, including their unique identifiers and categories. The connection matrix of the model is given by Table 2. This model was also used in [6], but now it is analyzed with the newer, improved version of the method. This way the results of the earlier and the improved methods are comparable, and the advantages of improvements become visible.

**Table 1.** Concept IDs, names and categories of the investigated model

Concept ID	Concept name	Category
C1	Clients	Assets
C2	Rules & regulations	
C3	New IT solutions	
C4	Funding	Money
C5	Cost reduction	
C6	Profit/loss	Financials
C7	Investments	
C8	Staff	Human resources
C9	New services	Product and process development
C10	Quality	
C11	Client development	Output measures
C12	Service development	
C13	Productivity	

#### 3.1 Properties of the original model

The properties of the original model was analyzed by simulations. The value of the  $\lambda$  parameter of FCM's threshold function was set to 5. Two FPs were found:

**Table 2.** Connection matrix of the FCM model

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
C1	0	0	0.5	0	0	0.5	1	0.5	0	0.5	1	0.5	0
C2	1	0	0.5	1	0	0	1	1	0.5	0	1	1	0
C3	1	0.5	0	0	0	-1	0	-1	1	0	1	1	1
C4	0	0	0	0	0	0	0	0	0	0	0	0	0
C5	0	0	1	-0.5	0	0	0	-1	0	0	0	1	0
C6	0	0	0	0	-0.5	0	0	0	0	0	0	0	0
C7	0.5	0	0.5	1	0	0.5	0	0	0	-0.5	0	0	0
C8	0	0	0	0	0	-0.5	0	0	0	0.5	0	0	-0.5
C9	0	0	0	1	0	0.5	0.5	0.5	0	-0.5	0	0.5	0
C10	0.5	0	0	0	0	0	0.5	0.5	0.5	0	1	0	0
C11	0	0	0.5	0.5	0	0	0	0	0.5	0.5	0	0	1
C12	0	0	0.5	0.5	0	0	1	0	0.5	0	0.5	0	-0.5
C13	0	0	1	0	0	0.5	0	0	0	0	1	0	0

23.1% of the 1000 investigated scenarios led to the first, and the remaining 76.9% to the second FP. Regardless of the investigated scenarios, most concepts had the same final values. Only the values of C6 and C8 differentiates the two FPs (see Table 3).

**Table 3.** Fixed-point attractors of the model

Concepts	C1-C3, C5, C7, C9-C13	C6	C8
FP#1	1.000	0.150	0.990
FP#2	1.000	0.855	0.922

### 3.2 Results of the analysis

Table 4 and Table 5 show the connection matrices of the two best bacteria of the last generation. Both of them were located on the first Pareto-front. Other bacteria are members of the other three Pareto-fronts. The most important properties of these model variants are collected in Table 6.

## 4 Conclusions

The improved method has reached its goal: it finds interesting model versions with smaller modifications than its preceding version while the modified models



**Table 4.** Connection matrix of the 1<sup>st</sup> model variant

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
C1	0.0	0.0	0.75	0.0	0.0	0.5	1.0	0.25	0.0	0.25	0.75	0.5	0.0
C2	1.0	0.0	0.5	1.0	0.0	0.0	0.5	0.75	0.75	0.0	0.75	1.0	0.0
C3	0.25	0.0	0.0	0.0	0.0	-1.0	0.0	-0.75	0.5	0.0	0.5	0.75	1.0
C4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C5	0.0	0.0	1.0	-0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0
C6	0.0	0.0	0.0	0.0	-0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C7	0.75	0.0	-0.25	1.0	0.0	0.25	0.0	0.0	0.0	1.0	0.0	0.0	0.0
C8	0.0	0.0	0.0	0.0	0.0	0.25	0.0	0.0	0.0	0.75	0.0	0.0	-0.5
C9	0.0	0.0	0.0	1.0	0.0	0.25	-0.75	0.25	0.0	0.25	0.0	0.75	0.0
C10	-0.75	0.0	0.0	0.0	0.0	0.0	0.25	-0.5	0.25	0.0	-0.25	0.0	0.0
C11	0.0	0.0	0.75	0.5	0.0	0.0	0.0	0.0	0.5	0.5	0.0	0.0	1.0
C12	0.0	0.0	0.5	0.5	0.0	0.0	0.25	0.0	0.25	0.0	-0.75	0.0	-0.5
C13	0.0	0.0	0.75	0.0	0.0	0.75	0.0	0.0	0.0	0.0	0.75	0.0	0.0

**Table 5.** Connection matrix of the 2<sup>nd</sup> model variant

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
C1	0.0	0.0	0.5	0.0	0.0	0.5	1.0	1.0	0.0	1.0	0.75	0.5	0.0
C2	-0.75	0.0	0.0	1.0	0.0	0.0	0.5	0.0	0.5	0.0	0.0	0.5	0.0
C3	0.25	0.25	0.0	0.0	0.0	-0.75	0.0	-1.0	0.0	0.0	0.5	-0.5	0.0
C4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C5	0.0	0.0	-0.5	-0.5	0.0	0.0	0.0	0.25	0.0	0.0	0.0	0.5	0.0
C6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C7	0.5	0.0	0.0	1.0	0.0	0.75	0.0	0.0	0.0	-0.5	0.0	0.0	0.0
C8	0.0	0.0	0.0	0.0	0.0	-1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.5
C9	0.0	0.0	0.0	1.0	0.0	0.0	0.75	1.0	0.0	-0.75	0.0	0.75	0.0
C10	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.5	-0.25	0.0	-0.75	0.0	0.0
C11	0.0	0.0	-1.0	0.5	0.0	0.0	0.0	0.0	0.25	1.0	0.0	0.0	-1.0
C12	0.0	0.0	0.0	0.5	0.0	0.0	1.0	0.0	0.5	0.0	-0.25	0.0	0.25
C13	0.0	0.0	1.0	0.0	0.0	0.75	0.0	0.0	0.0	0.0	0.5	0.0	0.0

**Table 6.** Main properties of the modified model variants

Property	1 <sup>st</sup> variant	2 <sup>nd</sup> variant
$\lambda$ value	2.366	2.070
Number of FPs	44	48
Number of LCs	0	0
Number of chaotic behavior	0	0
Difference from orig. model ( $d$ )	15.938	30.500

still have much more FPs than the original. There are several ways of possible further improvements, however.

The biggest obstacle to the application of the method is its performance: due to the high number of executed simulations, the process is extremely time consuming. BEA could be obviously accelerated: the parallel execution of mutations could be done trivially, but even the parallel version of gene transfer is worked out [17]. The implementation of these techniques are the next tasks.

The analysis could be further accelerated by the selection of some interesting connections, and modify only these connections while preserve the weight of others. It also looks useful to limit the range of new weight values to a specified interval.

BEA is slightly modified in our program to find Pareto-optimal solutions. This aim could be achieved several ways, the different possible implementations should be thoroughly investigated.

*Acknowledgement.* This research was supported BY the ÚNKP-17-4 New National Excellence Program of the Ministry of Human Capacities.

## References

- [1] Busemeyer, J.R.: Dynamic decision making (1999)
- [2] Salmeron, J.L.: Supporting decision makers with fuzzy cognitive maps. *Research-Technology Management* **52**(3) (2009) 53–59
- [3] Papageorgiou, E.I.: Fuzzy Cognitive Maps for Applied Sciences and Engineering: From Fundamentals to Extensions and Learning Algorithms. Volume 54. Springer Science & Business Media (2013)
- [4] Papageorgiou, E.I.: Learning algorithms for fuzzy cognitive maps—a review study. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **42**(2) (2012) 150–163
- [5] Hatwágner, M.F., Niskanen, V.A., Kóczy, L.T.: Behavioral analysis of fuzzy cognitive map models by simulation. In: Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS), 2017 Joint 17th World Congress of International, IEEE (2017) 1–6
- [6] Miklós F. Hatwágner, Gyula Vastag, V.A.N., Kóczy, L.T.: Banking applications of fcm models. In: 9th European Symposium on Computational Intelligence and Mathematics, <http://escim2017.uca.es/wp-content/uploads/2015/02/OralCommunications.pdf> (2017) 60–68
- [7] Deb, K.: Multi-objective optimization using evolutionary algorithms. Volume 16. John Wiley & Sons (2001)
- [8] Hatwágner, M.F., Kóczy, L.T.: Parameterization and concept optimization of fcm models. In: Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on, IEEE (2015) 1–8
- [9] Axelrod, R.: Structure of decision: The cognitive maps of political elites. Princeton university press (1976)
- [10] Kosko, B.: Fuzzy cognitive maps. *International Journal of man-machine studies* **24**(1) (1986) 65–75

- [11] Stylios, C.D., Groumpos, P.P.: Mathematical formulation of fuzzy cognitive maps. In: Proceedings of the 7th Mediterranean Conference on Control and Automation. (1999) 2251–2261
- [12] Tsadiras, A.K.: Comparing the inference capabilities of binary, trivalent and sigmoid fuzzy cognitive maps. *Information Sciences* **178**(20) (2008) 3880–3894
- [13] Nawa, N.E., Furuhashi, T.: A study on the effect of transfer of genes for the bacterial evolutionary algorithm. In: Knowledge-Based Intelligent Electronic Systems, 1998. Proceedings KES'98. 1998 Second International Conference on. Volume 3., IEEE (1998) 585–590
- [14] Nawa, N.E., Furuhashi, T.: Fuzzy system parameters discovery by bacterial evolutionary algorithm. *IEEE Transactions on Fuzzy Systems* **7**(5) (1999) 608–616
- [15] Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **28**(1) (1979) 100–108
- [16] Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63**(2) (2001) 411–423
- [17] Hatwagner, M., Horvath, A.: Parallel gene transfer operations for the bacterial evolutionary algorithm. *Acta Technica Jaurinensis* **4**(1) (2011) 89–111